Accessing accurate documents by mining auxiliary document information

Jinju Joby P.
Department of Computer Science and Engineering
Christ University Faculty of Engineering
Bangalore, India

Jyothi Korra
Computer Science and Engineering
Christ University Faculty of Engineering
Bangalore, India
jyothi.korra@christuniversity.in

Abstract

Earlier techniques of text mining included algorithms like k-means, Nave Bayes, SVM which classify and cluster the text document for mining relevant information about the documents. The need for improving the mining techniques has us searching for techniques using the available algorithms. This paper proposes one technique which uses the auxiliary information that is present inside the text documents to improve the mining. This auxiliary information can be a description to the content. This information can be either useful or completely useless for mining. The user should assess the worth of the auxiliary information before considering this technique for text mining. In this paper, a combination of classical clustering algorithms is used to mine the datasets. The algorithm runs in two stages which carry out mining at different levels of abstraction. The clustered documents would then be classified based on the necessary groups. The proposed technique is aimed at improved results of document clustering.

1 Introduction

Text mining has reached greater levels as continuous research has been moving on to find newer and improved techniques to mine text data. The researchers have been using the basic mining algorithms and techniques such as K-means, agglomerative hierarchical clustering and such to find the best text mining results. The combinations of the techniques also have made quite an impressive mark in data mining like bisect-K-means [1]. The commonly used clustering techniques like scatter/gather has been beneficial to understand how clustering happens in a real web world scenario [2]. The available tools for text mining are helpful as most of them are open-source and have a wide range of options to work with. Some of the commercially available tools like SPSS are common to data miners [3]. Apart from all these advances, researches are on an ever-growing thirst to find better ways to overcome the data mining difficulties as the data of the present world provides us with a range of parameters which cannot be well assessed using the existing tools and techniques. In such a search text mining on documents also require attention. The most important asset anywhere is data. To maintain the quality of data we go to extends of providing ways in which data can be managed, stored and used efficiently. As the amount of data increases exponentially as we speak, better ways to manage it is on high demand. Algorithms and techniques that do the same are to be found and deployed. Text mining on documents has been done for years. Ways to mine the text and cluster the documents for better processing is our concern. The document mining algorithm that

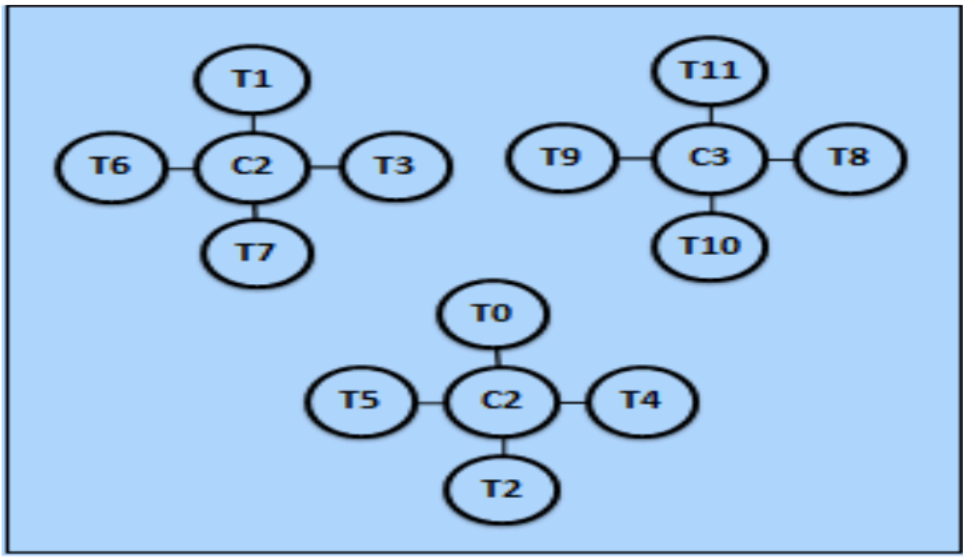
is proposed in this paper is a combination of few of the best text clustering algorithms and other data mining techniques. After the documents have been clustered into the most appropriate clusters, they are then classified into classes under which they belong most appropriately. The use of such document mining techniques can be applied in dataset management, to maintain data quality. The proposed techniques can be applied in order to cluster and classify the large number of documents that are otherwise disorganized. This is mainly required for easy access to the accurate document in minimum time. Thus, improving the mining techniques that can be used in the ever growing size of documents collected.

2 Related Work

The concept of document clustering has been a hot area of for research since the time data collection has taken birth. Researchers have been developing best and improved text mining algorithms and techniques since years. Even today as the size of data increases exponentially, text mining is a very important research area. This need has led to the proposal of an improved technique to mine the document dataset in order to cluster and classify the documents and thus to retrieve the most accurate document that is related to the query. The documents which are to be clustered are collected from a source which would be the World Wide Web or a repository which contains the different documents all in the same format [4][5]. The issue of clustering these documents into a way which would trigger better retrieval should be developed. The raw form of these documents is just too scattered and would take a user hours or even days to search for the required document. Since processing time is very expensive in application it is the basic need to be clustered and classified for ease of use. The algorithms for text clustering and classification like Nave Bayes, K-means, and SVM with kernerls [6, 7, 8, **?**] have been used to mine textual data [9]. Nave Bayes theorem was previously used to develop a system which detects the offensive content on the World Wide Web [10]. The messages posted on the user walls were passed through the mining algorithm which runs based on the concept of Nave Bayes. The resultant set of messages was classified based on the content and thus the messages with the offensive content would be blocked. The K-means algorithm is used strictly for clustering. The content based clustering requires an elaborate use of techniques along with K-means in order for the proposed methods to produce efficient results [1][11]. The different techniques that are used for clustering and classification were summarized by various authors [12][13][**?**]. A lot of comparison is done between the algorithms for document clustering like agglomerative and K-means. K- means algorithm is used as it gives efficient results and the K- means algorithm gives more superior results. Scatter/Gather technique involves the hybrid working model of both K-means and agglomerative hierarchical clustering. The bisecting K- means also produces results of clustered documents that are as good as those developed by using the agglomerative hierarchical clustering[1]. K-means is also applied after reducing the dimensions using PCA[14]. The approach of using auxiliary information that is available along with the documents for the mining of documents is also a well-known approach. The web logs, provenance information that is attached along with the document as the side information. This technique would allow better text mining as the documents would be clustered into better clusters. The K-means algorithm is used to cluster the document into the initial clusters which are just vague clustering. Then the similarity measures are taken in order to cluster the documents better. Later the auxiliary information is made use of to improve the clustering and thus the resultant clustering of the algorithm would give us the refined clusters. This technique is extended to classify the documents and we would obtain clusters which will allow the fast retrieval of a query asking for a document from the large dataset.

3 DOCUMENT CLUSTERING WITH AUXILARY INFORMATION

The documents used in the clustering process should all contain auxiliary information. The second part of the proposed method as discussed in section IV would be used based on the auxiliary information. The first stage of clustering requires the documents to be clustered first based on K-means algorithms. The similarity between the documents is found using cosine similarity. The cluster centroids will be updated based on the similarity values. Now to improve the clustering process, we need to remove the noisy attributes which will be useless for the clustering. This can be done by calculating the Gini index of the attributes in the documents.



The set of useful attributes would be used for clustering thus refining the clustering process. Now the attributes and the document clusters have been selected. Now the proposed model would need to combine both these features in order to find the final clusters which will allow easier retrieval of documents. This can be done by finding the posterior probability of the documents with respect to the attributes and the clusters. The process would require the documents to have similar attributes so that the clustering and the calculation of Gini index can be easily dealt with. The designed algorithm that is explained in section IV would put light on how the scattered documents would be clustered and these clustered can be further classified into classes. These classes would be then labelled accordingly so that the system would be able to retrieve the accurate documents in the least execution time. Thus the documents that are clustered and classified based on the attributes or auxiliary information is found to return better results than the normal document clustering algorithms and techniques. Certain preprocessing techniques like removal of stop words and if the required attributes can be found then it should be arranged in the required format.

4 PROPOSED METHOD

In this section, we discuss the proposed method for the design on a clustering and classification technique for fast retrieval of documents from a large dataset. The entire process has been broken down for better grasping of the approach.

4.1 PreprocessingDocuments

The documents that are collected from the World Wide Web or repositories need not always have the same formatting. Also if the auxiliary information is known but is not incorporated in the document itself then it should be done. The auxiliary attributes should be distinguished from the rest of the content by appending a special character in front of it like a $ or a #. This way the algorithm can distinguish between the actual content of the document and the auxiliary information. This is important because different parts of the designed algorithm are used on the content and the auxiliary information separately. It is also well studied that information in web page is viewed in the order of position of words[15].

4.2 Process Flow

The process flow of the clustering process occurs as shown in Fig 2. The documents that are used for clustering purposes are always represented as vectors. Each document d would be considered as a vector d, the document can be expressed as

$$df_t = tf_1 + tf_2 + ... + tf_n$$

, where dtf is the document and tfi is the frequency of the term i in the document. The corpus would be all the documents in the dataset represented by T. The K-means algorithm is used to randomly select k documents as the cluster centroids and then the remaining documents are placed in any of the k clusters randomly. This way the first step of clustering is done. Now in order to find if the documents in the cluster are indeed correlated to each other, we need to check the similarity measure of the documents with the centroid of the cluster. The cosine similarity can be used for this checking. The similarity between document Ti and the cluster centroid Ci will determine if the document indeed belongs to that cluster or not. This way the cosine similarity of all the documents with all the centroids should be checked in order to make sure that the documents are present in the cluster with maximum relevance and is not misplaced. As we are still in the K-means algorithm, the next set of cluster centroids have to be calculated according to the new clusters formed. These clusters would be index on each document to mark them as clustered. The part of clustering which involves only the documents is now done. Next we consider the auxiliary information or attribute that is present in the document. For example for the book ?In the year 2889?, the auxiliary information would be,
Title: In the Year 2889
Author: Jules Verne and Michel Verne
Release Date: January 2, 2007
Language: English
Character set encoding: ASCII.
The Gini index of each attribute r in the cluster Ci is found. The Gini index is calculated by

$$Pr_j = \frac{Fr_j}{\sum_{k=1}^{n} Fr_k}$$

, where Prj is the presence on attribute r in cluster j which is determined by the fraction of attribute r in cluster j, frj. If the attribute r is present in the cluster j then its value is 1. The summation of all the Prj of the attribute r in all the clusters is the Gini index of the attribute. Higher the value of Gini index, more useful the attribute is. As we may consider useless attributes for the clustering process and this may reduce the effectiveness of the proposed algorithm, the useless attributes need to be trimmed away. Thus now we mark only those attributes that have Gini index above a threshold value as usable attributes. This will now contain only the set of attributes Ri that are absolutely useful for the clustering purpose. The correlation between the documents in the cluster and the most relevant attributes in the cluster is used to find the final cluster. This can be done by finding the posterior probability between the attributes and the documents in the cluster using,

$$P(T_i \in C_i | R_i)$$

, which describes the posterior probability of attribute set Ri in the document Ti which is in the cluster Ci. The posterior probability of the document in each cluster is calculated and the highest value would be the cluster to which it belongs finally. This is the best cluster assignment that can be done based on the auxiliary attributes to the document.

4.3 Document Classification

This section is an extension of the clustering process where the clusters that are obtained would be classified. In order to classify the clusters we require the knowledge of the type of documents that are present in each cluster. Classification accuracy is verified using crowdsourcing approach [16]. The documents that are clustered into a single cluster must contain maximum number of common attributes. This is because the clustering has been done based on only the most relevant attributes. Hence if the comparison is done between all the attributes in the cluster the type of majority documents could be obtained. This can also mean that the same cluster can be classified in many ways depending on the attributes present. One way of classification is by finding the most repeating attribute in the cluster and naming the cluster by the attribute. Another way of clustering is by calculating the presence of all the attributes in the cluster and the cluster will be named after the attribute whose presence would have the highest value. Presence of an attribute is calculated by the formula described in subsection B. The classification process flow is as shown in Fig 4. After classification the documents can belong to different classes because the different attributes can influence the classification as shown in Fig 3.

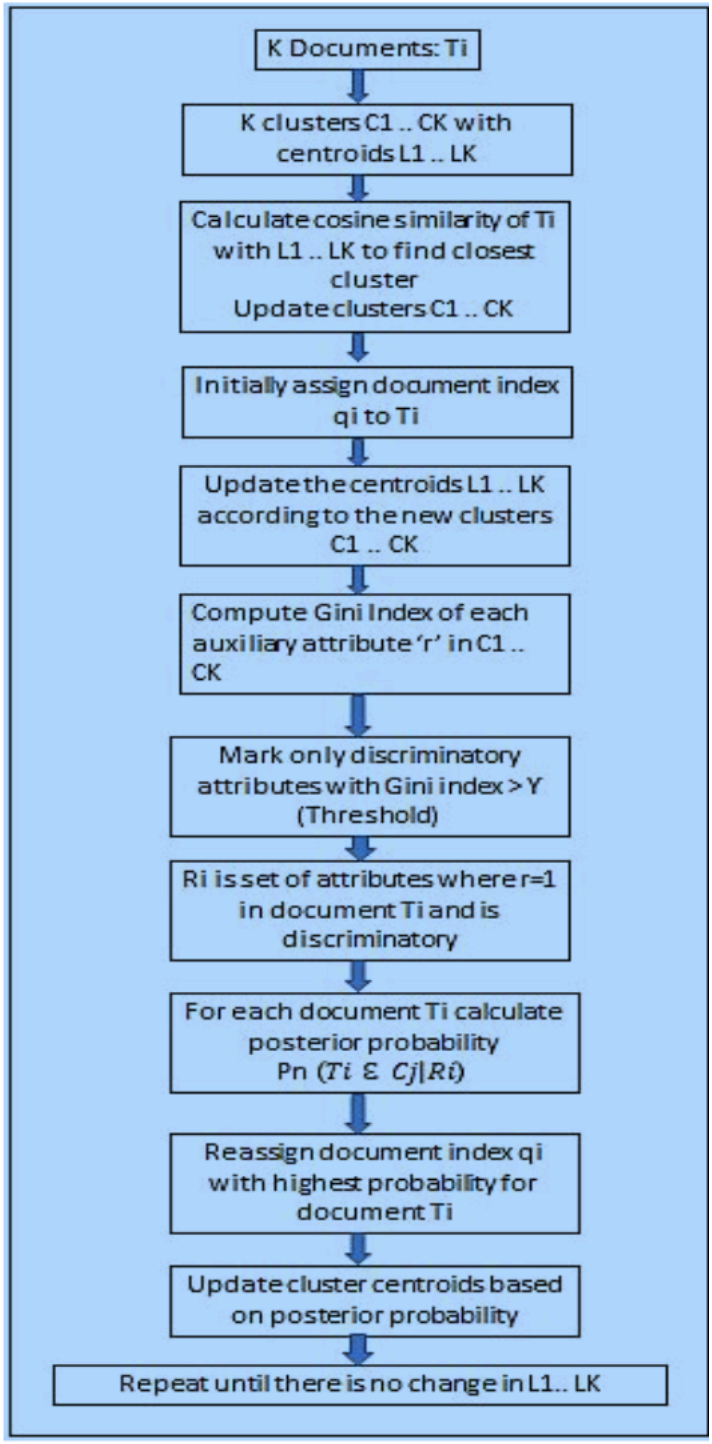


Fig 2. Proposed Clustering Model

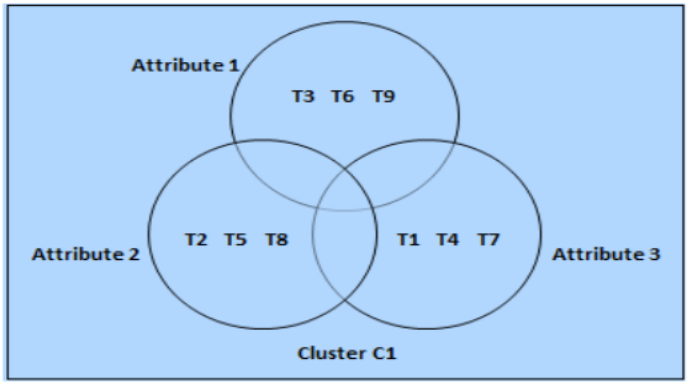


Fig 3. Classified Documents based on attributes

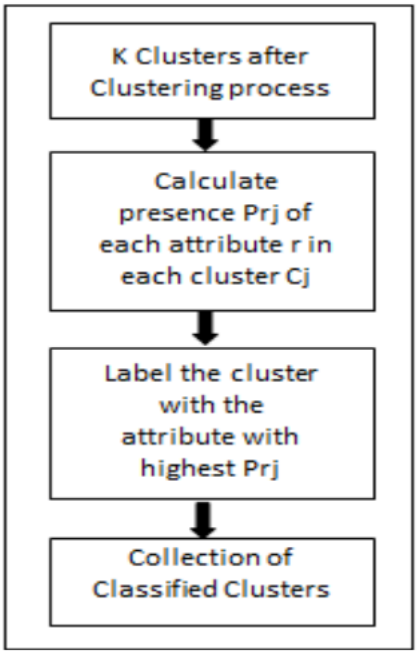


Fig 4. Proposed Classification Model

TABLE I. Results of clustering done on variety of datasets

Cluster ID	Size	Cosine Similarity	Knowledge discovery	Web mining	Privacy	Information technology
C1	972	0.081	948	24		
C2	948	0.075	81	13	61	792
C3	988	0.077	5	2		
C4	713	0.053	46	388	272	
C5	528	0.051	3	11	107	2
C6	806	0.076	1		795	

Cluster ID	Size	Cosine Similarity	Programming	Logic	Mathematics	WWW
C1	972	0.081				
C2	948	0.075				1
C3	988	0.077	980		1	
C4	713	0.053	2			5
C5	528	0.051	1	318	86	
C6	806	0.076	1		9	

5 EXPERIMENTAL RESULTS

We have collected the datasets which contain a large set of documents. These documents are collected from a journal which contains the papers published in computer science and its applied areas. The algorithm is implemented into a system which takes the input as the dataset and the output would be in two stages. The correctly clustered set of documents also the classified clusters based on the highest attribute presence. Table I gives the details of the clustered documents and the cosine similarity values after the entire algorithm has processed the dataset. The above documents also have the attribute list as shown previously in section IV. The attributes would be useful to filter out the unnecessary steps of processing by removing the unnecessary and useless attributes. Thus the refined set of attributes only would be allowed for the further clustering of the documents. This is how the retrieval process becomes better when compared to the basic clustering algorithms and techniques. This clustering would contain the content and attribute based documents which would generate the most accurate documents for the query submitted by the user. This is the aim of the proposed algorithm.

6 CONCLUSION AND FUTURE WORK

The proposed technique is aimed at the retrieval and accessing the documents out of a large dataset using the auxiliary attributes. The usual clustering process only uses the content in the documents to cluster. This paper proposes an approach to do the same, but by refining the clustering process using the attributes also. The auxiliary attributes like author details, paper details and Web logs would be considered as auxiliary attributes. After clustering, the classification can be done by using the next

step, where the most prominent attribute would be considered as the class of the cluster. This paper proposes this method in order to overcome the problem of maintaining and handling the large sets of data that we need to deal with.

References

[1] M. Steinbach, G. Karypis, V. Kumar *et al.*, "A comparison of document clustering techniques," in *KDD workshop on text mining*, vol. 400, no. 1. Boston, 2000, pp. 525–526.

[2] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey, "Scatter/gather: A cluster-based approach to browsing large document collections," in *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1992, pp. 318–329.

[3] M. Andronie, D. Crisan *et al.*, "Commercially available data mining tools used in the economic environment," *Database Systems Journal*, vol. 1, no. 2, pp. 45–54, 2010.

[4] S. Vijayarani and M. Muthulakshmi, "Comparative analysis of bayes and lazy classification algorithms," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 8, pp. 3118–3124, 2013.

[5] N. J. Belkin and W. B. Croft, "Information filtering and information retrieval: Two sides of the same coin?" *Communications of the ACM*, vol. 35, no. 12, pp. 29–38, 1992.

[6] S. N. Jagarlapudi, D. G, R. S, C. Bhattacharyya, A. Ben-tal, and R. K.r., "On the algorithmics and applications of a mixed-norm based kernel learning formulation," pp. 844–852, 2009. [Online]. Available: http://papers.nips.cc/paper/3880-on-the-algorithmics-and-applications-of-a-mixed-norm-based-kernel-learning-formulation.pdf

[7] D. Govindaraj, S. Raman, S. Menon, and C. Bhattacharyya, "Controlled sparsity kernel learning," *CoRR*, vol. abs/1401.0116, 2014. [Online]. Available: http://arxiv.org/abs/1401.0116

[8] P. Ravipally and D. Govindaraj, "Sparse classifier design based on the shapley value," in *Proceedings of the World Congress on Engineering*, vol. 1, 2010.

[9] Y. Zhao, G. Karypis, and U. Fayyad, "Hierarchical clustering algorithms for document datasets," *Data mining and knowledge discovery*, vol. 10, no. 2, pp. 141–168, 2005.

[10] M. Vanetti, E. Binaghi, E. Ferrari, B. Carminati, and M. Carullo, "A system to filter unwanted messages from osn user walls," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, no. 2, pp. 285–297, 2013.

[11] C. C. Aggarwal, S. C. Gates, and P. S. Yu, "On the merits of building categorization systems by supervised clustering," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1999, pp. 352–356.

[12] M. M. V. Upasani and R. C. Samant, "A review on meta information based text data clustering." *International Journal of Computer Science and Information Technologies*, vol. 5, no. 6, 2014.

[13] S. S. Raut and V. Maral, "Text clustering and classification on the use of side information," *International Journal of Science and Research (IJSR)*, vol. 3, no. 10, 2014.

[14] D. Govindaraj, "Application of active appearance model to automatic face replacement," *Journal of Applied Statistics*, 2011.

[15] D. Govindaraj, T. Wang, and S. V. N. Vishwanathan, "Modeling attractiveness and multiple clicks in sponsored search results," *CoRR*, vol. abs/1401.0255, 2014. [Online]. Available: http://arxiv.org/abs/1401.0255

[16] D. Govindaraj, N. K.V.M., A. Nandi, G. Narlikar, and V. Poosala, "Moneybee: Towards enabling a ubiquitous, efficient, and easy-to-use mobile crowdsourcing service in the emerging market," *Bell Labs Technical Journal*, vol. 15, no. 4, pp. 79–92, 2011. [Online]. Available: http://dx.doi.org/10.1002/bltj.20473